Claims:

1.   A table management device using degraded data as
address data, wherein the degraded data is produced from input
data having a predetermined number of bits, the degraded data
having a smaller number of bits, the table management device
5   comprising:

a plurality of tables allowed to be concurrently
accessed according to the degraded data, wherein each of the
tables is allowed to register a predetermined number of pieces
of data, each of the pieces of data having a number of bits equal
10   to the predetermined number of bits of the input data;

a plurality of comparators provided for respective
ones of the plurality of tables, wherein each of the comparators
compares the input data to a piece of data read from a
corresponding table according to the degraded data to produce
15   a comparison result; and

a determiner for determining from comparison
results of the comparators whether the input data has been
already registered in the tables.

2.   The table management device according to claim 1,
20   further comprising:

a controller controlling the tables such that, when
the input data has not been registered in the tables and an

available memory area exists in memory space of the tables
concurrently accessed according to the degraded data, the input
data is registered as new data in the available memory area of
the tables.

5   3.   The table management device according to claim 1,
wherein

              each of the comparators compares a corresponding
piece of data to the input data to produce a comparison result
indicating one of match and mismatch; and

10   the determiner determines that the input data has
been registered in the tables when a match-indicating comparison
result is received from at least one of the comparators, and
determines that the input data is not registered in the tables
when a mismatch-indicating comparison result is received from
15   each of the comparators.

          4.   The table management device according to claim 2,
wherein

              each of the comparators compares a corresponding
piece of data to the input data to produce a comparison result
20   indicating one of match and mismatch; and

              the determiner determines that the input data has
been registered in the tables when a match-indicating comparison
result is received from at least one of the comparators, and
determines that the input data is not registered in the tables

when a mismatch-indicating comparison result is received from each of the comparators.

5.   A method for managing an address table which is divided into a plurality of banks, comprising the steps of:

5   a) converting input address data having a predetermined number of bits to address data having a smaller number of bits according to hash processing;

b) concurrently accessing the banks according to the address data to read registered address data from each of the 10   banks;

c) comparing the input address data to the registered address data read from each of the banks to produce comparison results for respective ones of the banks; and

d) determining from the comparison results whether 15   the input address data has been registered in the address table.

6.   The method according to claim 5, further comprising the steps of:

e) when the input address data has not been registered in the address table, determining whether an 20   available memory area exists in memory space of the banks concurrently accessed according to the address data;

f) when an available memory area exists, registering the input address data as new address data in the available memory area; and

g) when no available memory area exists, changing
the hash processing.

7.   The method according to claim 6, wherein, in the step
(a), the hash processing is performed by selecting a
5   desired number of bits at a predetermined position from 32-
bit data obtained by CRC32 calculation.

8.   The method according to claim 7, wherein, in the step
(g), the hash processing is changed by selecting the
desired number of bits at another position different from the
10   predetermined position from the 32-bit data obtained by the
CRC32 calculation.

9.   A table management device using a hash output value
as an address, wherein the hash output value is obtained by
converting an input MAC (media access control) address based
15   on a hash function, the table management device comprising:

a MAC address table which is divided into a plurality
of banks allowed to be concurrently accessed using the hash
output value as an address;

a plurality of comparators provided for respective
20   ones of the plurality of banks, wherein each of the comparators
compares the input MAC address to a registered MAC address read
from a corresponding bank according to the hash output value
to produce a comparison result; and

a determiner for determining from comparison results of the comparators whether the input MAC address has been already registered in the MAC address table.

10.   The table management device according to claim 9, further comprising:

a controller controlling the MAC address table such that, when the input MAC address has not been registered in the MAC address table and an available memory area exists in memory space of the MAC address table accessed according to the hash output value, the input MAC address is registered as a new MAC address in the available memory area of the MAC address table.

11.   The table management device according to claim 9, wherein

each of the comparators compares a corresponding registered MAC address to the input MAC address to produce a comparison result indicating one of match and mismatch; and

the determiner determines that the input MAC address has been registered in the MAC address table when a match-indicating comparison result is received from at least one of the comparators, and determines that the input MAC address is not registered in the MAC address table when a mismatch-indicating comparison result is received from each of the comparators.

12.   A method for managing a plurality of tables,

comprising the steps of:

a) converting input data having a predetermined number of bits to degraded data having a smaller number of bits according to a predetermined algorithm;

b) concurrently accessing the plurality of tables according to the degraded data to read registered data from each of the tables;

c) comparing the input data to the registered data read from each of the banks to produce comparison results for respective ones of the tables; and

d) determining from the comparison results whether the input data has been registered in the tables.

13.   The method according to claim 12, further comprising the steps of:

e) when the input data has not been registered in the tables, determining whether an available memory area exists in memory space of the tables concurrently accessed according to the degraded data; and

f) when an available memory area exists, registering the input data as new data in the available memory area.

14.   The method according to claim 13, further comprising the step of:

g) when no available memory area exists, changing the predetermined algorithm.

15.   A computer program for use in a computer to manage a plurality of tables, the program comprising the steps of:

a) converting input data having a predetermined number of bits to degraded data having a smaller number of bits according to a predetermined algorithm;

b) concurrently accessing the plurality of tables according to the degraded data to read registered data from each of the tables;

c) comparing the input data to the registered data read from each of the banks to produce comparison results for respective ones of the tables; and

d) determining from the comparison results whether the input data has been registered in the tables.

16.   The computer program according to claim 15, further comprising the steps of:

e) when the input data has not been registered in the tables, determining whether an available memory area exists in memory space of the tables concurrently accessed according to the degraded data; and

f) when an available memory area exists, registering the input data as new data in the available memory area.

17.   The computer program according to claim 16, further comprising the step of:

g) when no available memory area exists, changing the predetermined algorithm.

18. A computer system comprising:

a plurality of tables; and

a processor on which a program is to run,

wherein the program comprises the steps of:

a) converting input data having a predetermined number of bits to degraded data having a smaller number of bits according to a predetermined algorithm;

b) concurrently accessing the plurality of tables according to the degraded data to read registered data from each of the tables;

c) comparing the input data to the registered data read from each of the banks to produce comparison results for respective ones of the tables; and

d) determining from the comparison results whether the input data has been registered in the tables.

19. The computer system according to claim 18, wherein the program further comprises the steps of:

e) when the input data has not been registered in the tables, determining whether an available memory area exists in memory space of the tables concurrently accessed according to the degraded data; and

f) when an available memory area exists, registering

the input data as new data in the available memory area.

        20.   The computer system according to claim 19, wherein
the program further comprises the step of:

             g) when no available memory area exists, changing
5   the predetermined algorithm.

        21.   A recording medium storing a computer program for
instructing a computer to manage a plurality of tables, the
program comprising the steps of:

             a) converting input data having a
10   predetermined number of bits to degraded data having a
smaller number of bits according to a predetermined algorithm;

             b) concurrently accessing the plurality of tables
according to the degraded data to read registered data from each
of the tables;

15   c) comparing the input data to the registered data
read from each of the banks to produce comparison results for
respective ones of the tables; and

             d) determining from the comparison results whether
the input data has been registered in the tables.